



 \bowtie **Flex** Go

HTML

Java **JavaScript**

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML



Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

R Bug (9) Code Smell 61 6 Vulnerability 5 All rules (76) Security Hotspot 1

Tags

Security.allowDomain(...) should only be used in a tightly focused manner Vulnerability flash.system.Security.exactSettings property should never be set to false Vulnerability Dynamic classes should not be used Code Smell "LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed Vulnerability "default" clauses should be first or last Code Smell Event types should be defined in metadata tags Code Smell Event names should not be hardcoded

in event listeners

Code Smell

The special "star" type should not be used

Code Smell

Variables of the "Object" type should not be used

Code Smell

Methods should not be empty

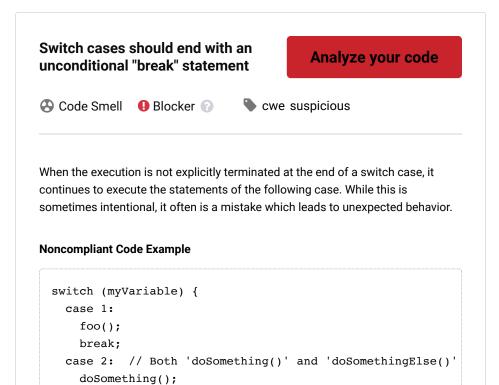
Code Smell

Constant names should comply with a naming convention

Code Smell

All branches in a conditional structure should not have exactly the same implementation

Bug



Search by name...

Compliant Solution

}

default:

break;

doSomethingElse();

```
switch (myVariable) {
  case 1:
    foo();
    break;
  case 2:
    doSomething();
    break:
  default:
    doSomethingElse();
    break;
}
```

Exceptions

This rule is relaxed in the following cases:

```
switch (myVariable) {
  case 0: // Empty case used to specify the same behavior f
  case 1:
    doSomething();
    break;
  case 2: // Use of return statement
    return;
  case 3: // Use of throw statement
    throw new IllegalStateException();
  case 4: // Use of continue statement
    continue:
  default: // For the last case, use of break statement is
    doSomethingElse();
}
```

See

- MITRE, CWE-484 Omitted Break Statement in Switch
- CERT, MSC17-C. Finish every set of statements associated with a case label

Classes that extend "Event" should override "Event.clone()" Rug Bug Constructors should not dispatch events 📆 Bug "ManagedEvents" tags should have companion "Event" tags 📆 Bug Objects should not be instantiated inside a loop Code Smell Two branches in a conditional structure should not have exactly the same implementation Constructor bodies should be as lightweight as possible Code Smell Only "while", "do" and "for" statements should be labelled Code Smell Statements, operators and keywords specific to ActionScript 2 should not be used Code Smell "for" loop stop conditions should be

invariant

Code Smell

Unused function parameters should

with a break statement

• CERT, MSC52-J. - Finish every set of statements associated with a case label with a break statement

Available In:

sonarcloud 🔗 sonarqube

 $@\ 2008-2022\ Sonar Source\ S.A.,\ Switzerland.\ All\ content\ is\ copyright\ protected.\ SONAR,$ SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of $Sonar Source \ S.A. \ All \ other \ trademarks \ and \ copyrights \ are \ the \ property \ of \ their$ respective owners. All rights are expressly reserved. Privacy Policy